AD-A096 972 COLORADO STATE UNIV FORT COLLINS DEPT OF MATHEMATICS F/G 12/1
AN ADAPTIVE PIECEWISE CURVE-FITTING PACKAGE USING A LOOK-AHEAD --ETC(U)
1981 D C PLATT, G D TAYLOR F49620-79-C-0124
UNCLASSIFIED AFOSR-TR-81-0310 NL

1 OF 1
AD A
096972

END
DATE
FILMED
4-81
DTIC

AD A096972

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR-81-0310	2. GOVT ACCESSION NO. AD-A096 972	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN ADAPTIVE PIECEWISE CURVE-FITTING PACKAGE USING A LOOK-AHEAD STRATEGY		5. TYPE OF REPORT & PERIOD COVERED Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David D. C. Platt ~~and~~ G. D. Taylor		8. CONTRACT OR GRANT NUMBER(s) F49620-79-C-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS Colorado State University Department of Mathematics Fort Collins, Colorado 80523		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2304/A3
11. CONTROLLING OFFICE NAME AND ADDRESS AFOSR/NM Bldg. 410 Bolling AFB, D.C. 20332		12. REPORT DATE 1981
		13. NUMBER OF PAGES 27 pages
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DTIC ELECTE MAR 30 1981 S D A

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An adaptive curve fitting algorithm for one dimensional data sets is developed and tested. This algorithm is a modification of two earlier curve fitting packages using the least squares (l_2) or the l_1 approximation operator. This new algorithm performs a look ahead strategy in the "backing off" routine for locating the knots joining the polynomial approximating pieces.

DD FORM 1 JAN 73 1473

UNCLASSIFIED 407344

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DTIC FILE COPY

AFOSR-TR- 81-0310

AN ADAPTIVE PIECEWISE CURVE-FITTING PACKAGE USING A LOOK-AHEAD STRATEGY

by

David C. Platt

David C. Platt

Advisor:

G. D. Taylor

F49620-79-C-0124

In partial fulfillment of the requirements for

the Degree of Master of Science

Department of Mathematics

Spring, 1981

A

1. Introduction

This paper presents an adaptive curve fitting algorithm based upon a look-ahead strategy. This algorithm is a modification of two earlier curve fitting packages. In the first section a discussion of the capabilities of the original routines is given and in the second section the new adaptive algorithm is described. The last section of the paper presents some numerical testing results.

2. Original ℓ_1, ℓ_2 Packages

In [1], a FORTRAN program is given which computes a smooth piecewise polynomial approximation, p, to a finite one-dimensional data set $\{(x_i, y_i)\}_{i=1}^m$ using the least squares (ℓ_2) or the ℓ_1 approximation operator. In order to describe these algorithms it is convenient to use a functional notation. Thus, let $X = \{x_i\}_{i=1}^m$ and f be the function defined on X by $f(x_i) = y_i$, $i = 1, \ldots, m$. Let $a = \min\{x: x \in X\}$ and $b = \max\{x: x \in X\}$ and, for any function g defined on $Y \subset X$ we use the notation $\|g\|_Y$ to represent $\max\{|g(x)|: x \in Y\}$.

A user of the code given in [1] must supply certain input parameters beyond the data $\{(x_i, y_i)\}_{i=1}^m$. These additional parameters include N and SMTH, where N - 1 is the desired degree of each approximating polynomial piece and SMTH is the desired smoothness or number of continuous derivatives of the fit on [a, b]. The user must also input a value for the desired percent error, PE, which is used to calculate the tolerance, TOL, which the approximating pieces must meet. That is, each approximating polynomial piece must fit the data in its interval of definition with a maximum absolute pointwise error less than TOL. In addition, the parameter NPTS must be supplied. This parameter allows the addition of artificial data

points. In particular, for a sparse data set, the user can add additional data points by setting NPTS > 0. In this case, the subroutine LINEAR will add NPTS new points by linear interpolating between consecutive data points.

In this context, the ℓ_1 and ℓ_2 packages will calculate an approximation p to f and a set of points (knots) $\{t_i\}_{i=0}^{k} \subset X$ with $a=t_0 < t_1 < \ldots < t_{k-1} < t_k=b$ such that

a. p restricted to $[t_{i-1}, t_i]$ is a polynomial $p_i \in \pi_{N-1}$ = {q: q is a real algebraic polynomial of degree $\leq N - 1$}.

b. p has SMTH continuous derivatives.

c. $\|f - p\|_X \leq$ TOL.

The first step in the ℓ_1 and ℓ_2 packages is to determine the location of the first knot t_1. In order to do this, the codes find $\tilde{t}_1$, the largest point in X such that

a. $[a, \tilde{t}_1] \cap X$ contains at least max(2, N + 1) points, and

b. If p_1 is the best approximation to f from π_{N-1} on $S_1 = [a, \tilde{t}_1] \cap X$, then $\|f - p_1\|_{S_1} \leq$ TOL.

If $\tilde{t}_1$ = b, the algorithm successfully terminates with no interior knots. If no such $\tilde{t}_1$ exists, then the algorithm aborts and an appropriate error message is printed.

If t_1 exists, $a < \tilde{t}_1 < b$, and SMTH ≥ 1, then the right endpoint of the first polynomial piece is determined by "backing off" from $\tilde{t}_1$ to a suitable point in S_1 using the following procedure.

The ℓ = N - SMTH - 1 largest extreme points of the error function $f(x) - p_1(x)$ are selected from the interval S_1 and one of them will be used for the knot t_1. The knot is chosen in this manner because, in the continuous setting, if f is differentiable and ξ is an interior relative

extreme point of $f(x) - p_1(x)$, then $f'(\xi) - p_1'(\xi) = 0$ so that $f'(\xi) = p_1'(\xi)$. Thus, joining p_2 to p_1 at ξ smoothly will force p_2 to have the same direction as f at the knot and should damp out unwanted oscillations.

In order to decide which extreme point to use for t_1, the code first computes the values $f'(\xi_1)$, $f'(\xi_2)$, ..., $f'(\xi_\ell)$ where $f'(\xi_\nu)$ is an approximation for the "slope" of the data at ξ_ν determined by the centered quadratic interpolation of the data at ξ_ν and its two immediate neighbors. The code then chooses the largest ξ such that $|f'(\xi_\nu) - p_1'(\xi_\nu)| \leq$ TOL. If no such ξ_ν exists, then t_1 is chosen to be the largest ξ_ν at which $|f'(\xi_\nu)-p_1'(\xi_\nu)|$ attains its minimum. This procedure is repeated on the intervals $[t_1, b]$, ..., $[t_n, b]$ until $t_n = b$.

3. The New Algorithm

The original ℓ_1 and ℓ_2 packages approximated with polynomial pieces in standard form (eg., $p_i(x) = \sum_{n=1}^{N} c_n x^{n-1}$). In the new algorithm, the polynomial pieces are of the form $p_i(x) = \sum_{n=1}^{N} c_n (x - x^*)^{n-1}$ where x^* is the left endpoint of the subinterval on which $p_i(x)$ is defined. However, the main focus of this algorithm is a new "backing off" strategy. This new "backing off" strategy can be characterized as a look-ahead strategy.

In the first step, the code will attempt to fit the entire data set on $S = [a, b] \cap X$ with a polynomial $p \in \pi_{N-1}$ such that $\|f - p\|_S \leq$ TOL. If this is not possible, then the algorithm will find the largest element of X, $\tilde{t}_1$, such that the best approximation p_1 to f on $S_1 = [a, \tilde{t}_1] \cap X$ satisfies $\|f - p_1\|_{S_1} \leq$ TOL. This is done via a halving procedure.

Specifically, if it is found that the best approximation on [a, b] does not satisfy the tolerance criterion then the code attempts to fit the data on the first N + 1 points of X. If it is unable to satisfy the tolerance criterion on this set, the code aborts, printing an appropriate error message. If the error criterion is satisfied on this small set then the point $\tilde{t}_1$ is computed via "bisection" initialized with the right endpoint of this small set and b. The code will then attempt to fit the rest of the data using $\tilde{t}_1$ as the current left endpoint. Specifically, the best approximation p_2 to f on $S_2 = [\tilde{t}_1, b] \cap X$ subject to $p_2^{(j)}(\tilde{t}_1) = p_1^{(j)}(\tilde{t}_1)$, j = 0, 1, ..., NSMTH is calculated. If $\|f - p_2\|_{S_2} \leq$ TOL holds, then the code successfully terminates with $\tilde{t}_1$ taken as our knot t_1. If $\|f - p_2\|_{S_2} >$ TOL then the bisection procedure is invoked once again to find $\tilde{t}_2 \in X$ such that $\tilde{t}_2$ is the largest element of X for which the best approximation p_2 to f on $S_2 = [\tilde{t}_1, \tilde{t}_2] \cap X$ subject to $p_2^{(j)}(\tilde{t}_1) = p_1^{(j)}(\tilde{t}_1)$, j = 0, 1, ..., NSMTH satisfies $\|f - p_2\|_{S_2} \leq$ TOL. The code stores the current second subinterval $[\tilde{t}_1, \tilde{t}_2]$ and the polynomial piece p_2 as the current knots and the best approximation to date. The code will then "back off" one point from $\tilde{t}_1$ to $\hat{t}_1$, the point of X immediate proceding $\tilde{t}_1$. Then the above procedure is applied to $[a, \hat{t}_1]$. That is, $\hat{t}_2 \in X$ is calculated such that $\hat{t}_2$ is the largest element of X for which the best approximation $\hat{p}_2$ to f on $\hat{S}_2 = [\hat{t}_1, \hat{t}_2] \cap X$ subject to $\hat{p}_2^{(j)}(\hat{t}_1) = \hat{p}_1^{(j)}(\hat{t}_1)$, j = 0, ..., NSMTH satisfies $\|f - \hat{p}_2\|_{\hat{S}_2} \leq$ TOL. If $\hat{t}_2$ = b then the algorithm successfully terminates with the two polynomial pieces p_1 and $\hat{p}_2$ and the knot $\hat{t}_1$ as the desired fit. If $\hat{t}_2 \leq \tilde{t}_2$ occurs the p_1 and p_2 and the knot $\tilde{t}_1$ remains the current candidate for the first two pieces and common knot for the final fit. If $\hat{t}_1 > \tilde{t}_2$ occurs then p_2, $\tilde{t}_1$ and $\tilde{t}_2$ are replaced by

$\hat{p}_2$, $\hat{t}_1$ and $\hat{t}_2$, respectively, and the above backing off procedure is continued. At this point two options are available. In the first option, denoted as COMPUT I, the code will continue backing off, one point at a time, and store the current $\tilde{t}_1$, $\tilde{t}_2$ and the approximating polynomial piece, p_2, that approximates farthest to the right subject to the smoothness constraints and tolerance criterion as the current knots and best second approximation piece replacing any previous lesser attempts. If the code backs off two consecutive points without fitting farther to the right subject to the smoothness constraints and given tolerance, then the code will accept the current $\tilde{t}_1$, $\tilde{t}_2$ and p_2 as the knots and best approximation for the second subinterval and begin work on a third subinterval which has $\tilde{t}_2$ as its left endpoint prior to "backing off". In the second option, denoted as COMPUT II, the code will back off a fixed number of points in an attempt to improve the current approximation. This fixed number is the greatest integer less than 1/2 the number of points in $[a, \tilde{t}_1] \cap X$, $\tilde{t}_1$ = the initial value for $\tilde{t}_1$.

4. Numerical Results

In most cases, the original ℓ_1 and ℓ_2 packages and the modified algorithms presented here produced similar results. Precisely, in 68 out of 79 data sets tested with the ℓ_1 packages and in 64 out of 83 data sets tested with the ℓ_2 packages similar results were produced by the original and the new algorithms. Some examples illustrating this are shown in Figures 1-12 on six given data sets.

In our testing we found that the original ℓ_1 and ℓ_2 packages produced better results than the new versions in a small but significant number of

cases. We believe that this is primarily due to the fact that the new versions are designed with the objective of reducing the number of knots by extending the length of each subinterval as far as possible subject to the given smoothness and tolerance requirements. In doing this, the new methods try to be too perfect too soon and are unable to recover later on in the approximation. One could say that our methods suffer from greediness. An example of this behavior is given by the Titanium data tested with the ℓ_2 packages (Figures 15, 16). Here the new method causes piecewise interpolation to result. Note that this instability of the new algorithm is not found in the original algorithm. On the other hand, for a few data sets, such as the data set Test 50000 the new algorithm for the ℓ_2 package (Figures 13, 14) reduced the number of knots needed. Thus, we cannot claim that the original ℓ_1 and ℓ_2 packages will always produce better fits than the revised versions. Likewise, there were five data sets where the original ℓ_1 package is superior to the new ℓ_1 package. However, there were also six data sets where the new ℓ_1 package produced a better fit than the original ℓ_1 package. For example, the ABS(SIN(X)) data set with noise and bad points was tested using the ℓ_1 packages (Figures 17, 18). Here the original package attempted to fit some of the bad points, whereas the new version did not.

In the testing of these codes it was observed that the COMPUT I and COMPUT II options gave the same results for all 162 data sets tested. Our results may not be conclusive for all cases but we will assume the following statement is reasonable. In general, backing off until two consecutive attempts do not improve the fit subject to the smoothness constraints and given tolerance will suffice.

Further backing up will not allow the current approximating piece to extend farther to the right.

In summary, based upon the limited testing that we performed, it appears that the original ℓ_1 and ℓ_2 codes are preferable to the modified codes presented here. We believe this conclusion is warranted due to the fact that the original codes appear to be more robust than the new codes using the look-ahead strategy and that when both codes performed satisfactorily, they gave similar results.

As a final note we also wish to point out that our study also illustrates one problem facing a user of such software. Namely, at the outset there is no way to predict the effectiveness of a given data fitting code. A good illustration of this is given in the ℓ_1 and ℓ_2 fits of the Titanium data (Figures 19 and 20). Here the ℓ_1 code was successful whereas the ℓ_2 code was not. Since this particular data is not particularly unusual, it is somewhat surprising that these two codes should vary so widely on it. We believe that this is an indication of the need for more research on the many problems involved with data fitting.

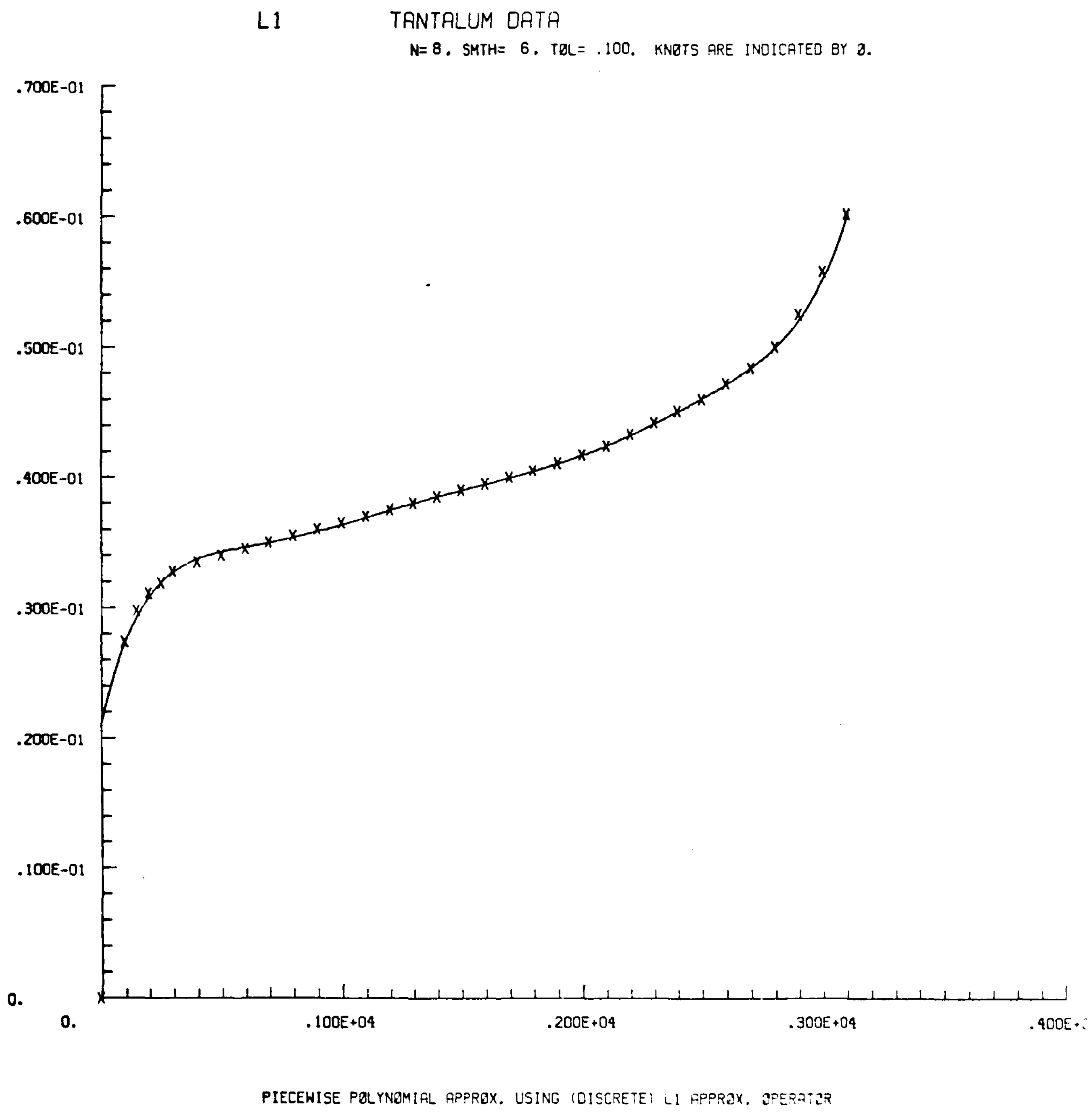

L1 TANTALUM DATA
N= 8, SMTH= 6, TØL= .100. KNØTS ARE INDICATED BY Ø.
.700E-01
.600E-01
.500E-01
.400E-01
.300E-01
.200E-01
.100E-01
0.
0.
.100E+04
.200E+04
.300E+04
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L1 APPRØX. OPERATOR

Figure 1

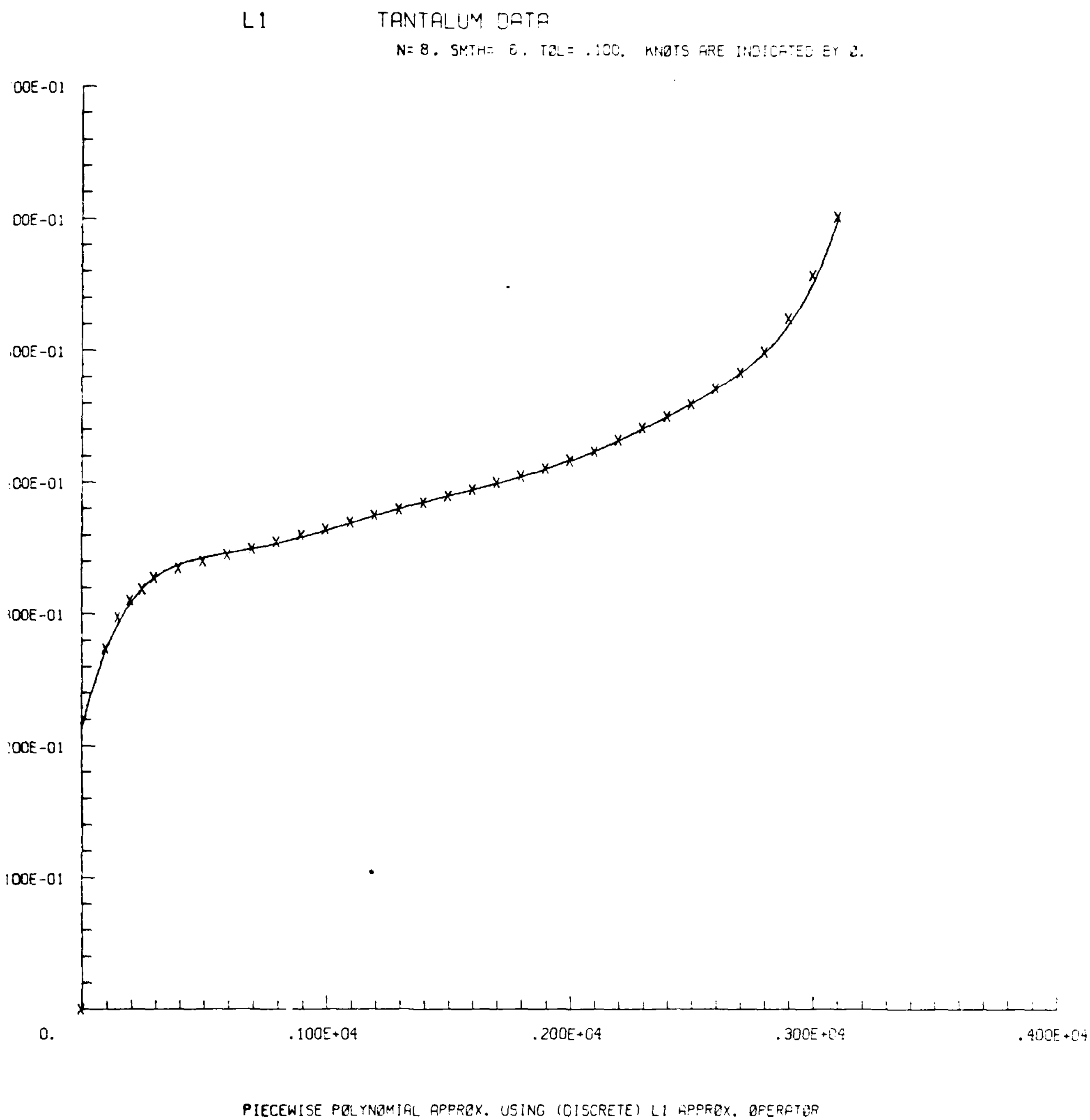

L1 TANTALUM DATA
N= 8. SMTH= 6. TØL= .100. KNØTS ARE INDICATED BY Ø.
00E-01
00E-01
00E-01
00E-01
00E-01
00E-01
00E-01
0.
.100E+04
.200E+04
.300E+04
.400E+04
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L1 APPRØX. ØPERATØR

Figure 2

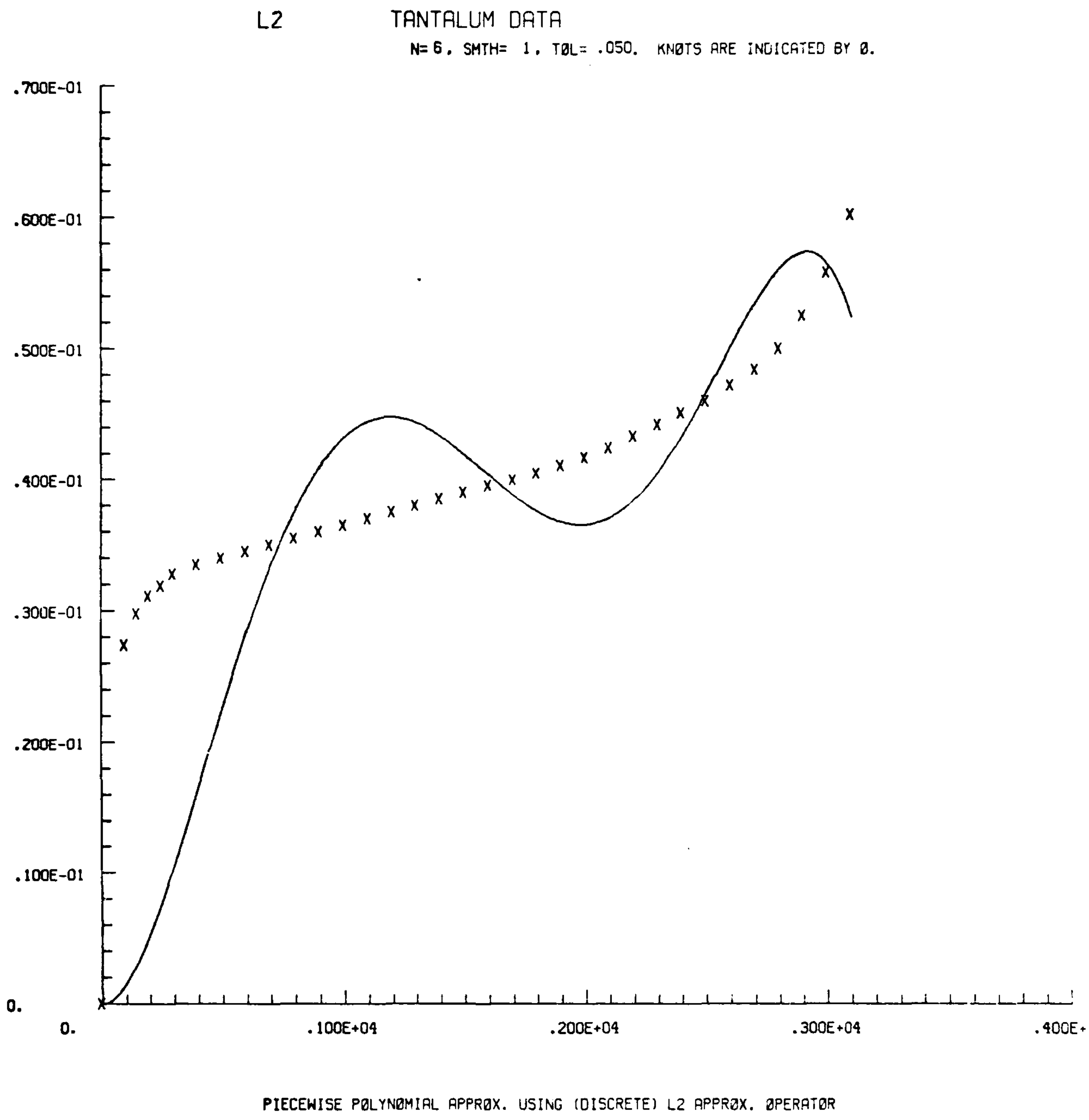

L2 TANTALUM DATA
N= 6, SMTH= 1, TØL= .050. KNØTS ARE INDICATED BY Ø.
.700E-01
.600E-01
.500E-01
.400E-01
.300E-01
.200E-01
.100E-01
0.
0.
.100E+04
.200E+04
.300E+04
.400E+
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L2 APPRØX. ØPERATØR

Figure 3

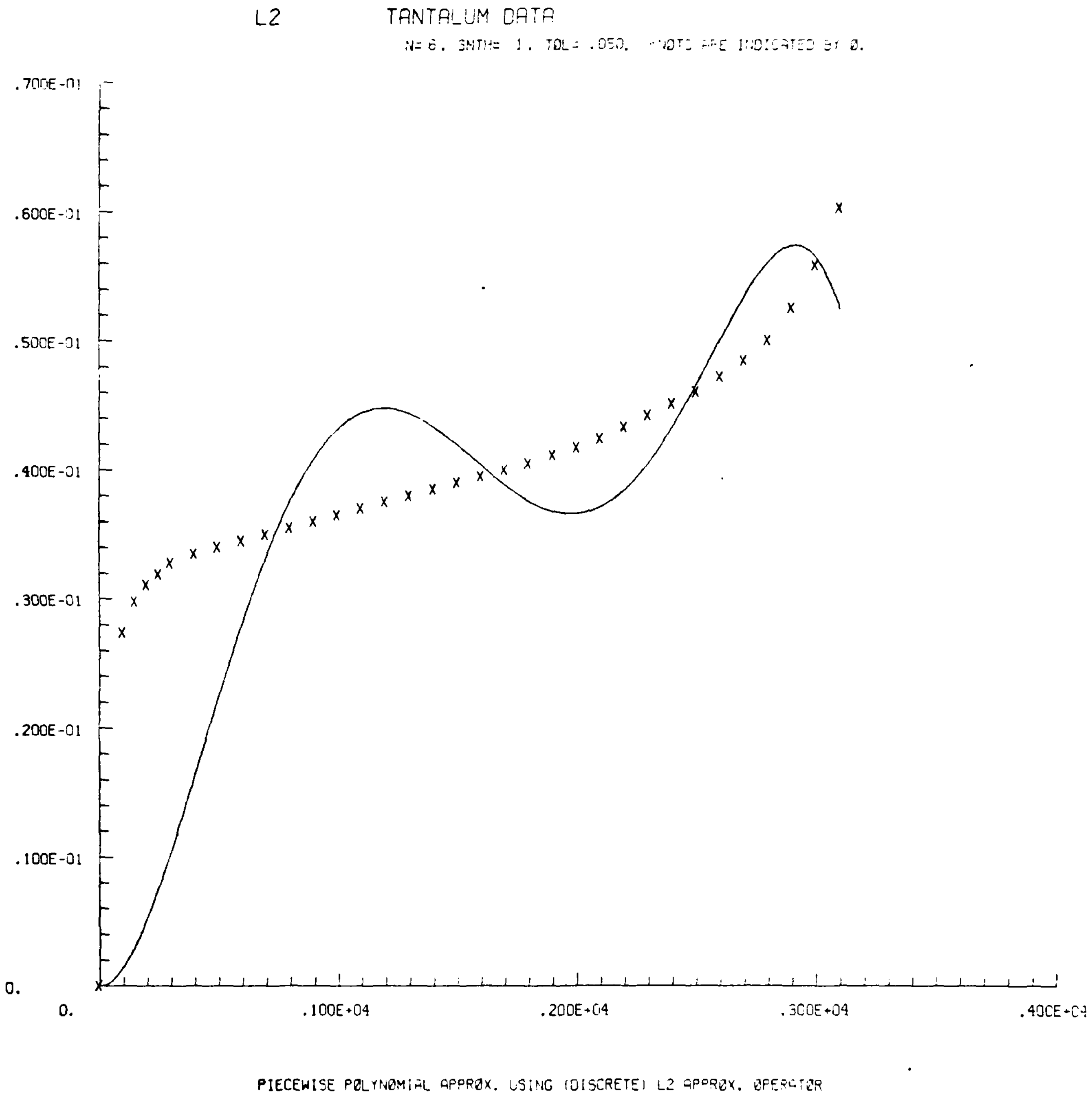

L2
TANTALUM DATA
N= 6. SNTH= 1. TOL= .050.
.700E-01
.600E-01
.500E-01
.400E-01
.300E-01
.200E-01
.100E-01
0.
0.
.100E+04
.200E+04
.300E+04
PIECEWISE POLYNOMIAL APPROX. USING (DISCRETE) L2 APPROX. OPERATOR

Figure 4

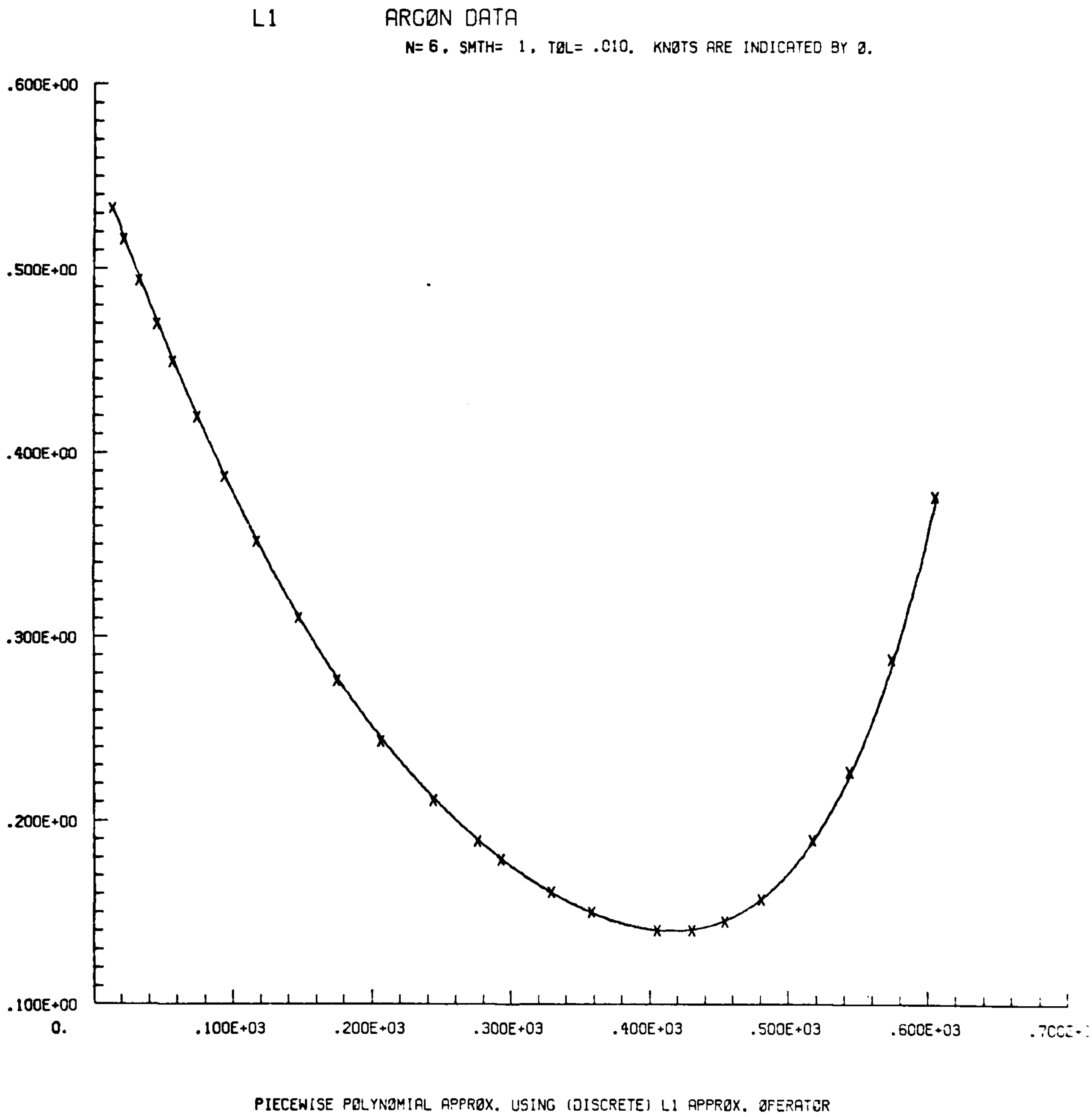

L1
ARGØN DATA
N= 6, SMTH= 1, TØL= .010, KNØTS ARE INDICATED BY Ø.
.600E+00
.500E+00
.400E+00
.300E+00
.200E+00
.100E+00
0.
.100E+03
.200E+03
.300E+03
.400E+03
.500E+03
.600E+03
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L1 APPRØX. ØPERATØR

Figure 5

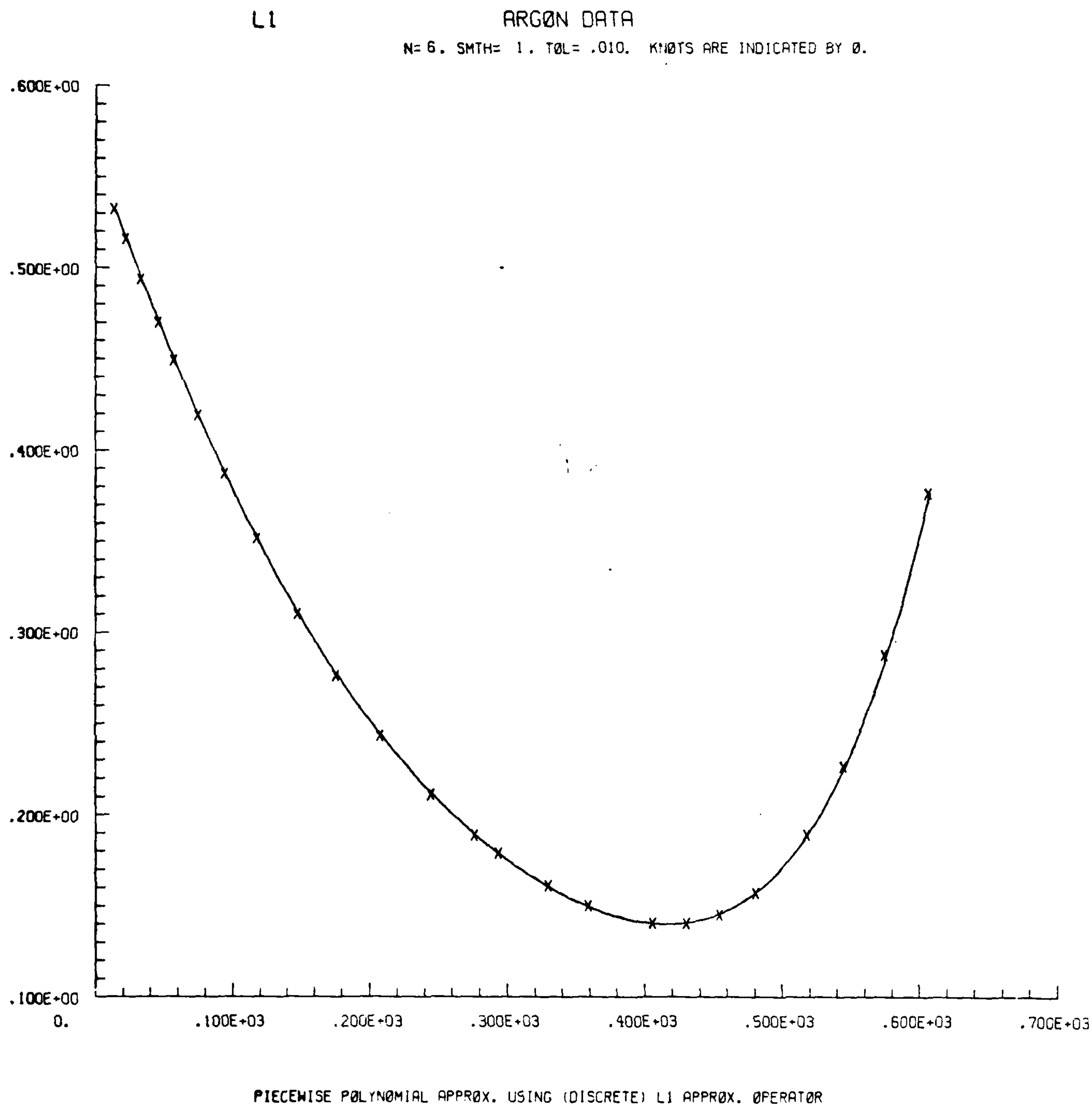
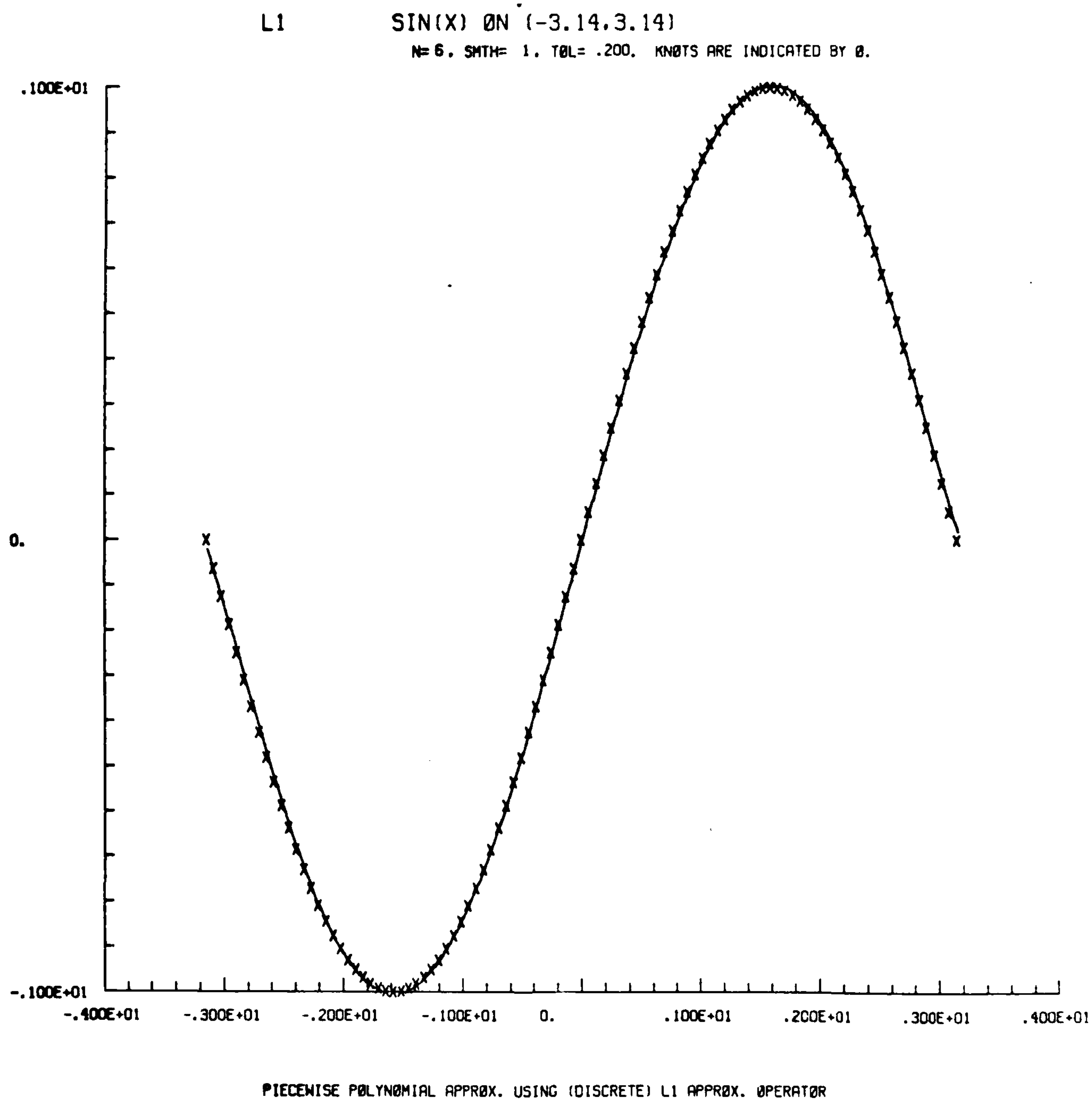
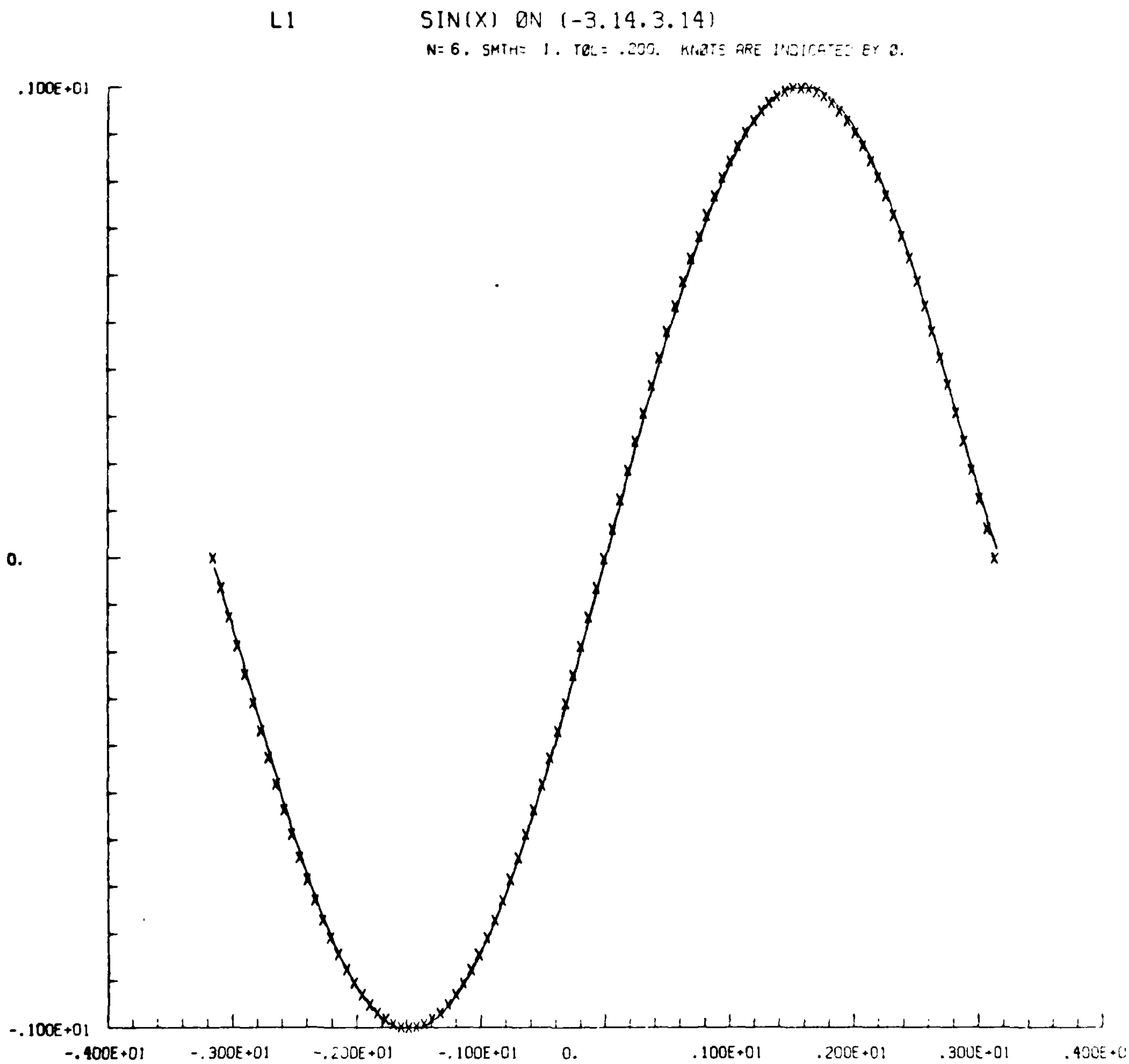

L1
ARGØN DATA
N= 6. SMTH= 1. TØL= .010. KNØTS ARE INDICATED BY Ø.
.600E+00
.500E+00
.400E+00
.300E+00
.200E+00
.100E+00
0.
.100E+03
.200E+03
.300E+03
.400E+03
.500E+03
.600E+03
.700E+03
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L1 APPRØX. ØPERATØR

Figure 6



L1 SIN(X) ØN (-3.14,3.14)
N= 6, SMTH= 1. TØL= .200. KNØTS ARE INDICATED BY Ø.
.100E+01
0.
-.100E+01
-.400E+01
-.300E+01
-.200E+01
-.100E+01
0.
.100E+01
.200E+01
.300E+01
.400E+01
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L1 APPRØX. ØPERATØR

Figure 7



PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L1 APPRØX. ØPERATØR

Figure 8

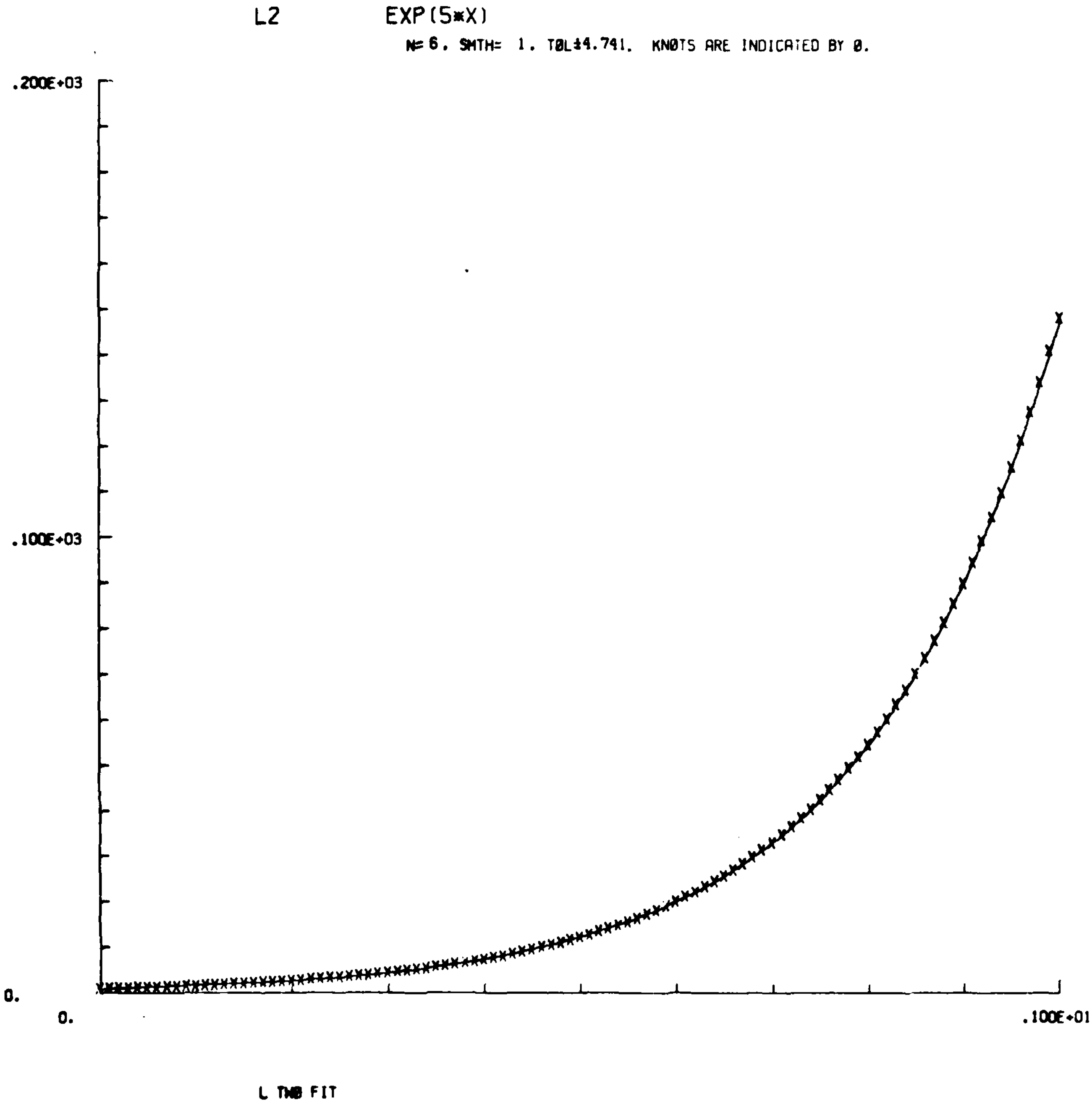

L2
EXP(5*X)
N=6, SMTH= 1, TØL=4.741, KNØTS ARE INDICATED BY Ø.
.200E+03
.100E+03
0.
0.
.100E+01
L TWØ FIT

Figure 9

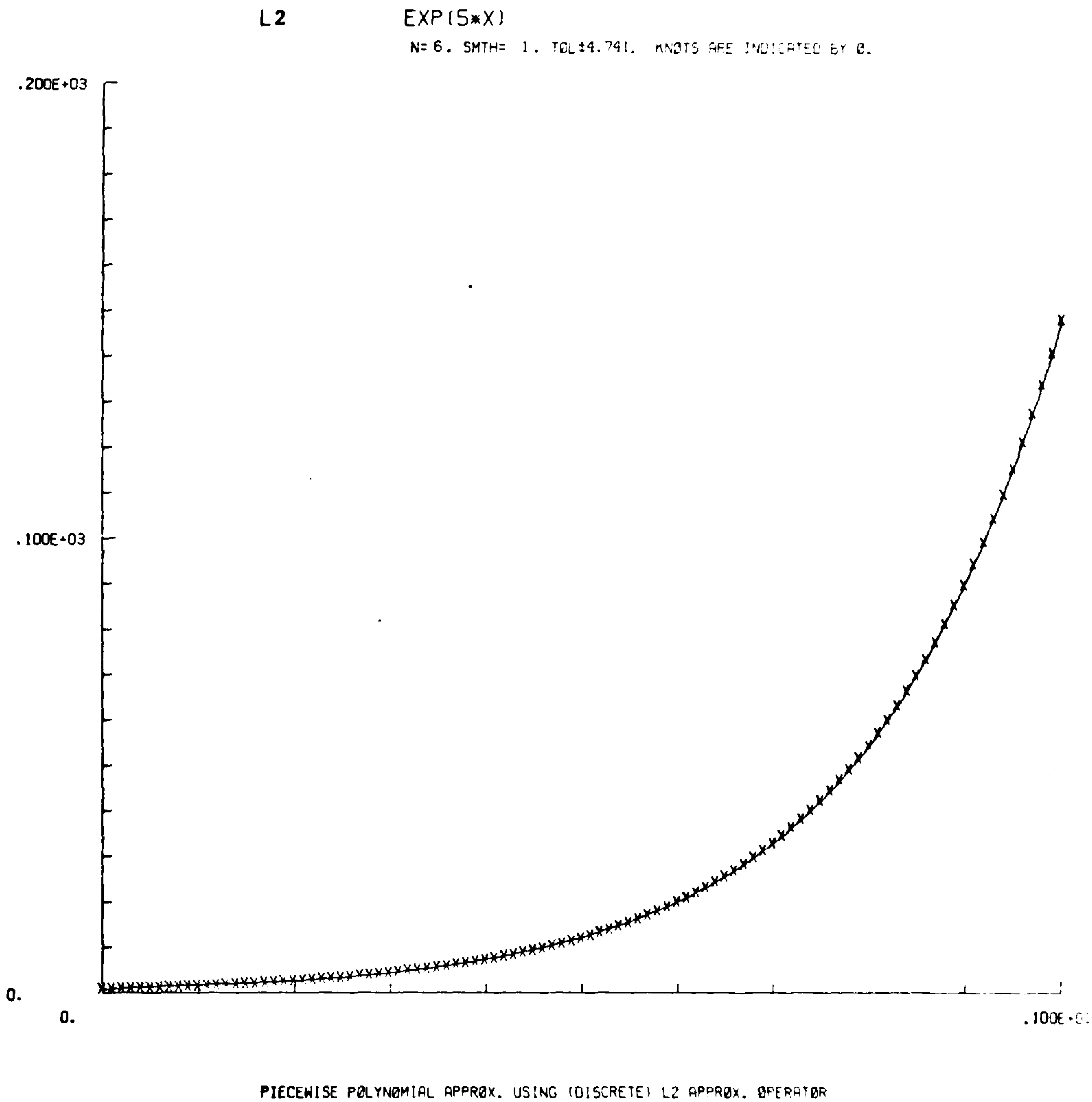
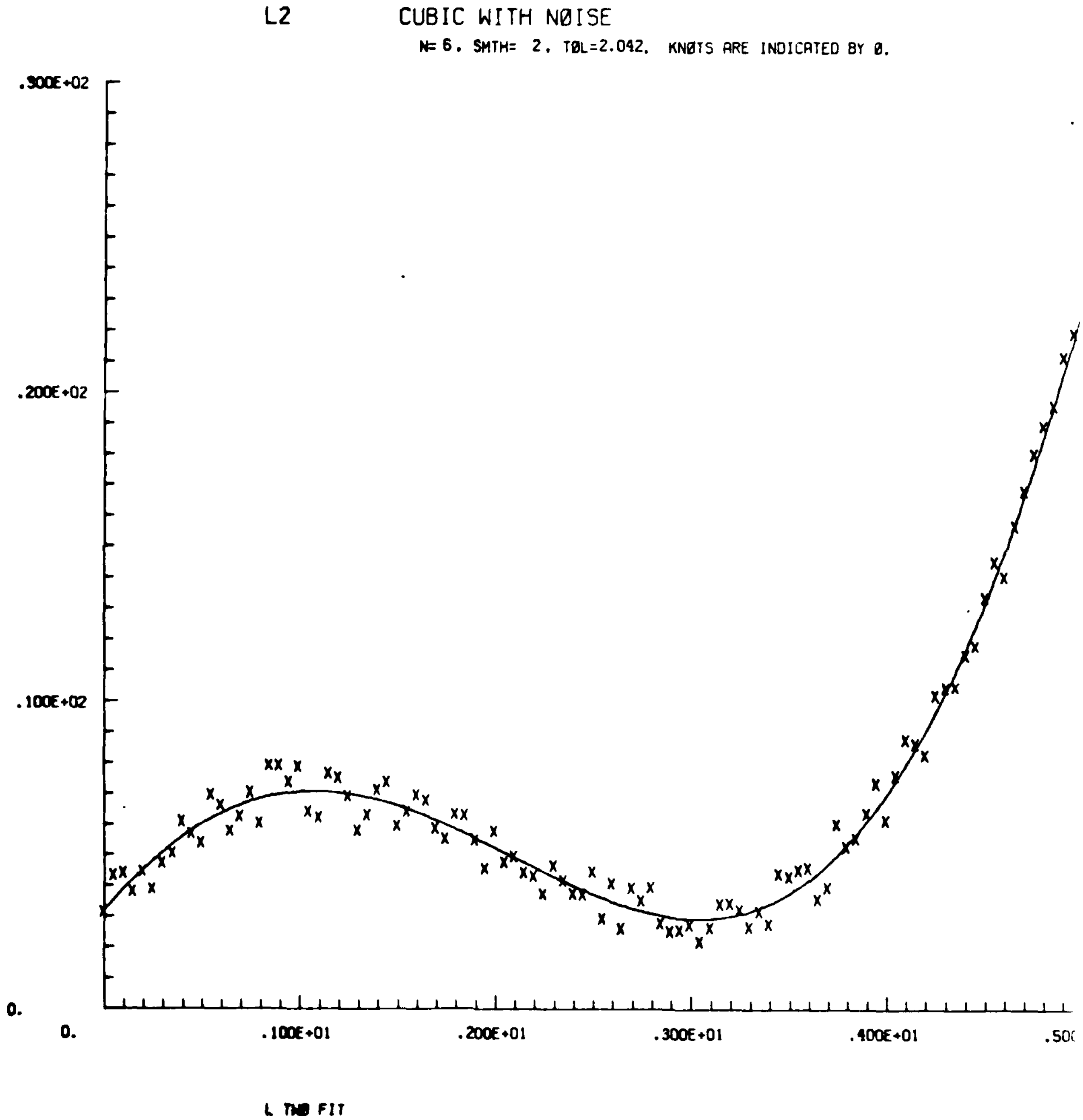
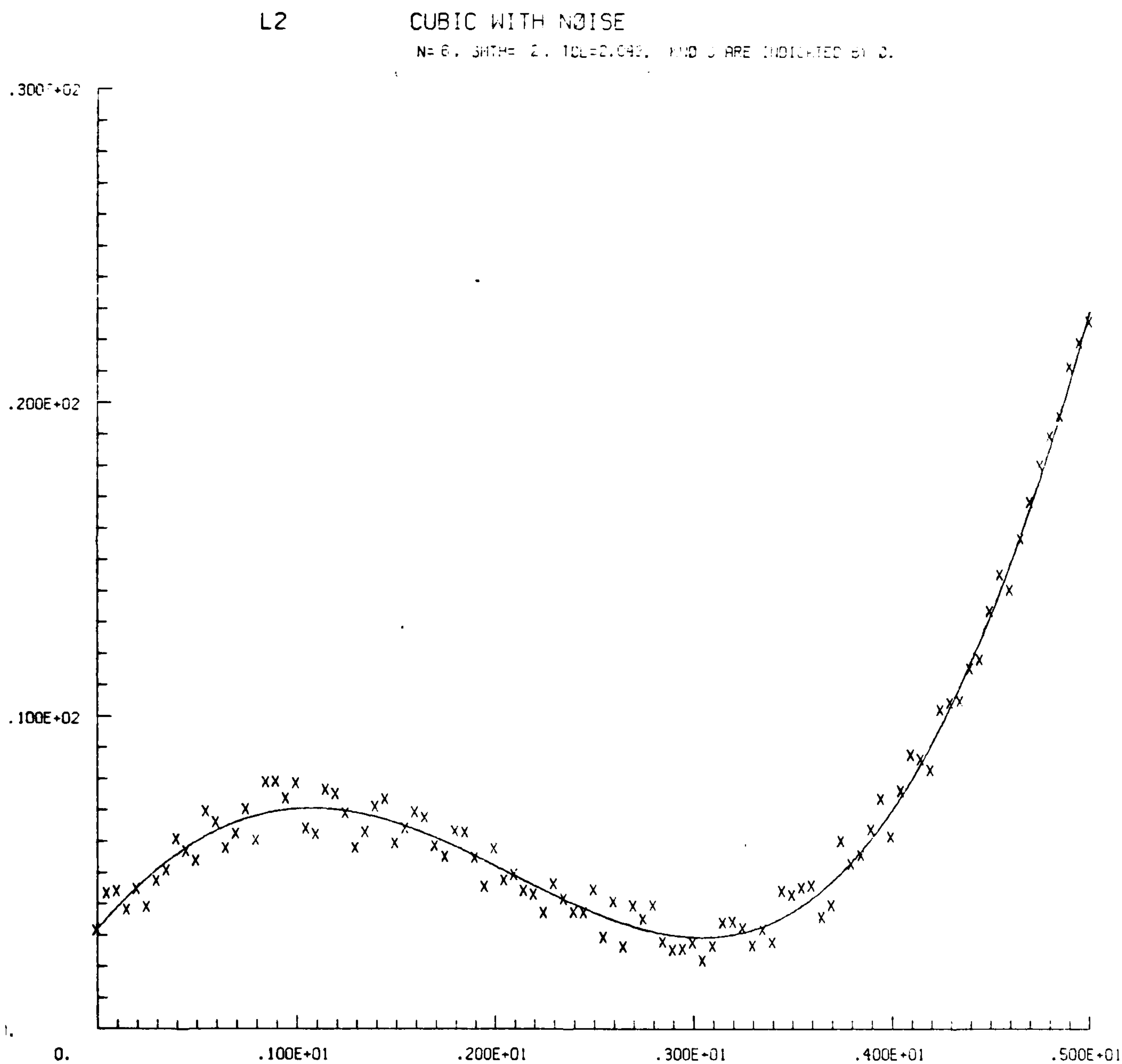

L2
EXP(5*X)
N= 6. SMTH= 1. TØL=4.741. KNØTS ARE INDICATED BY @.
.200E+03
.100E+03
0.
0.
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L2 APPRØX. ØPERATØR

Figure 10



L2 CUBIC WITH NØISE
N= 6, SMTH= 2, TØL=2.042, KNØTS ARE INDICATED BY Ø.
.300E+02
.200E+02
.100E+02
0.
0.
.100E+01
.200E+01
.300E+01
.400E+01
L TWØ FIT

Figure 11



PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L2 APPRØX. ØPERATØR

Figure 12

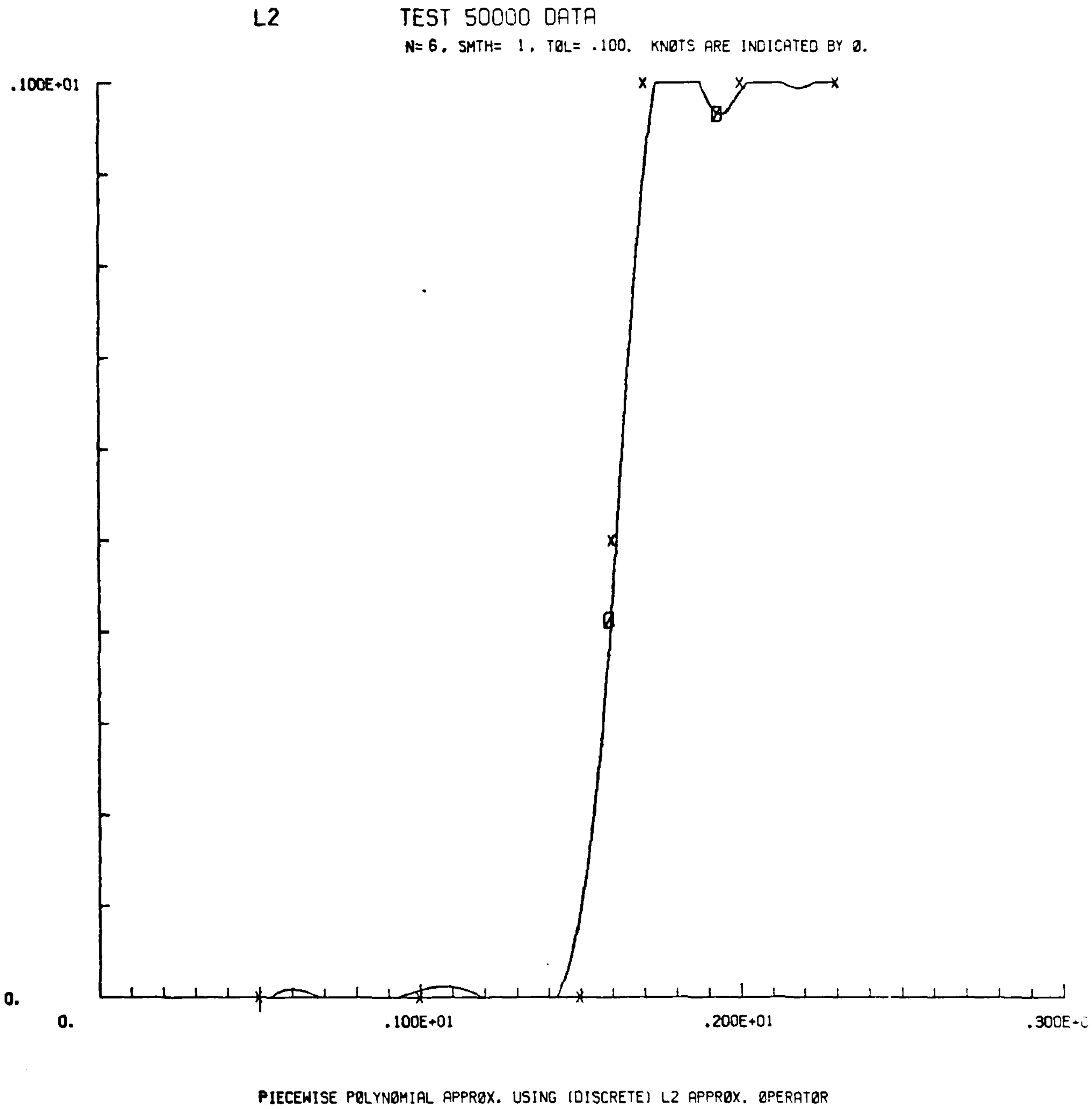


Figure 13

NPTS = 200

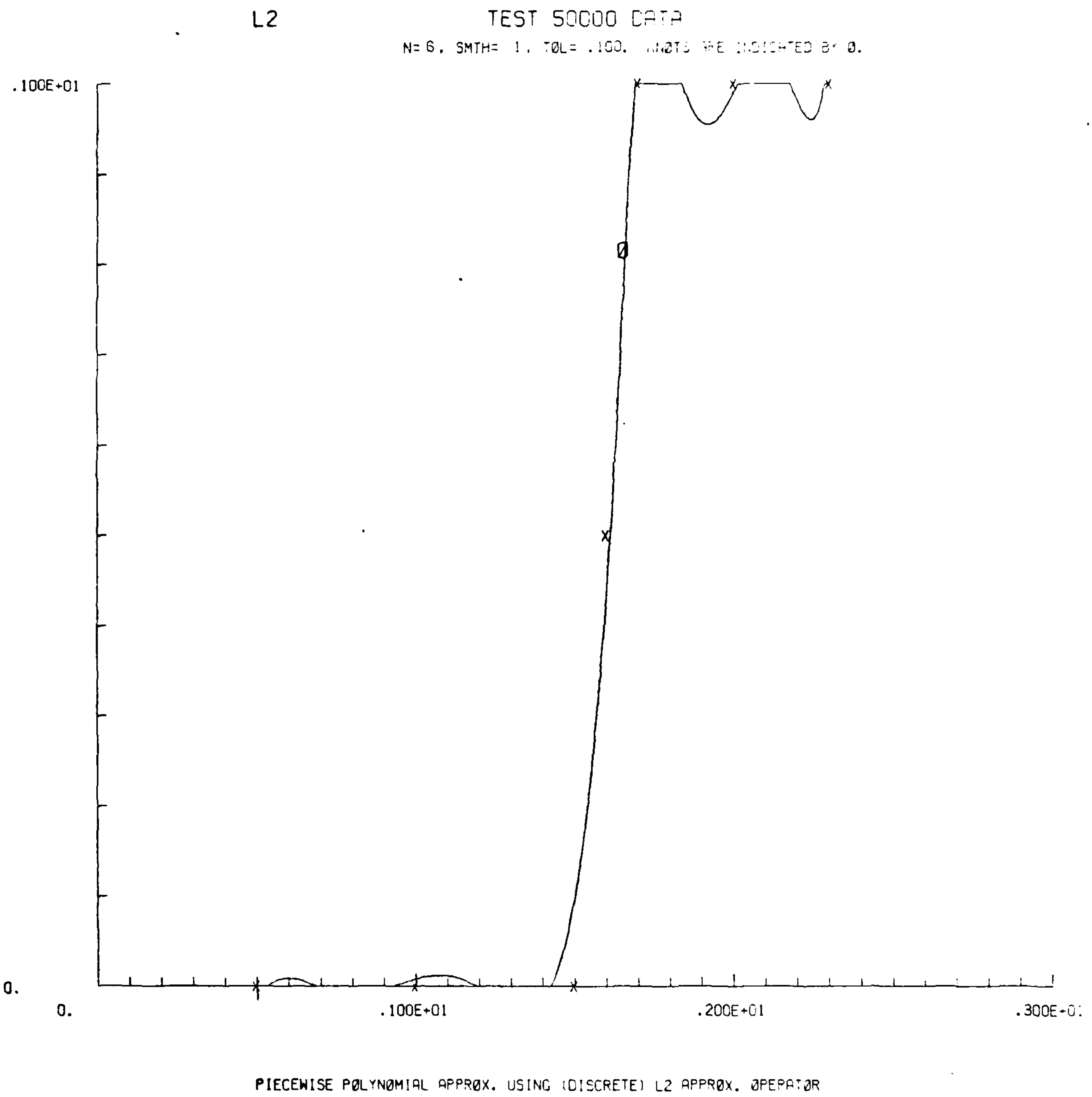

L2
TEST 50000 DATA
N= 6, SMTH= 1, TØL= .100.
.100E+01
0.
0.
.100E+01
.200E+01
Ø
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L2 APPRØX. ØPERATØR

Figure 14

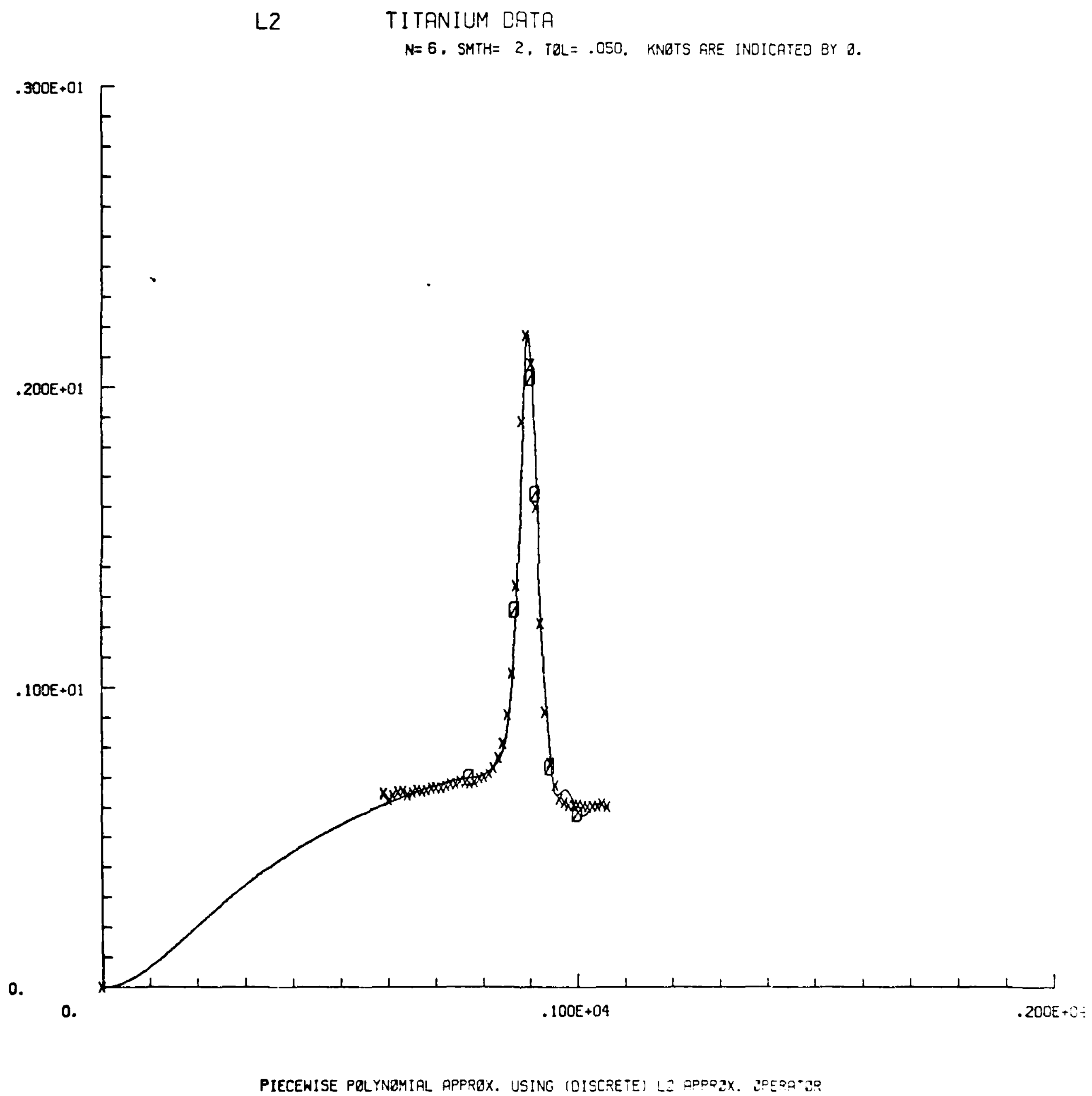
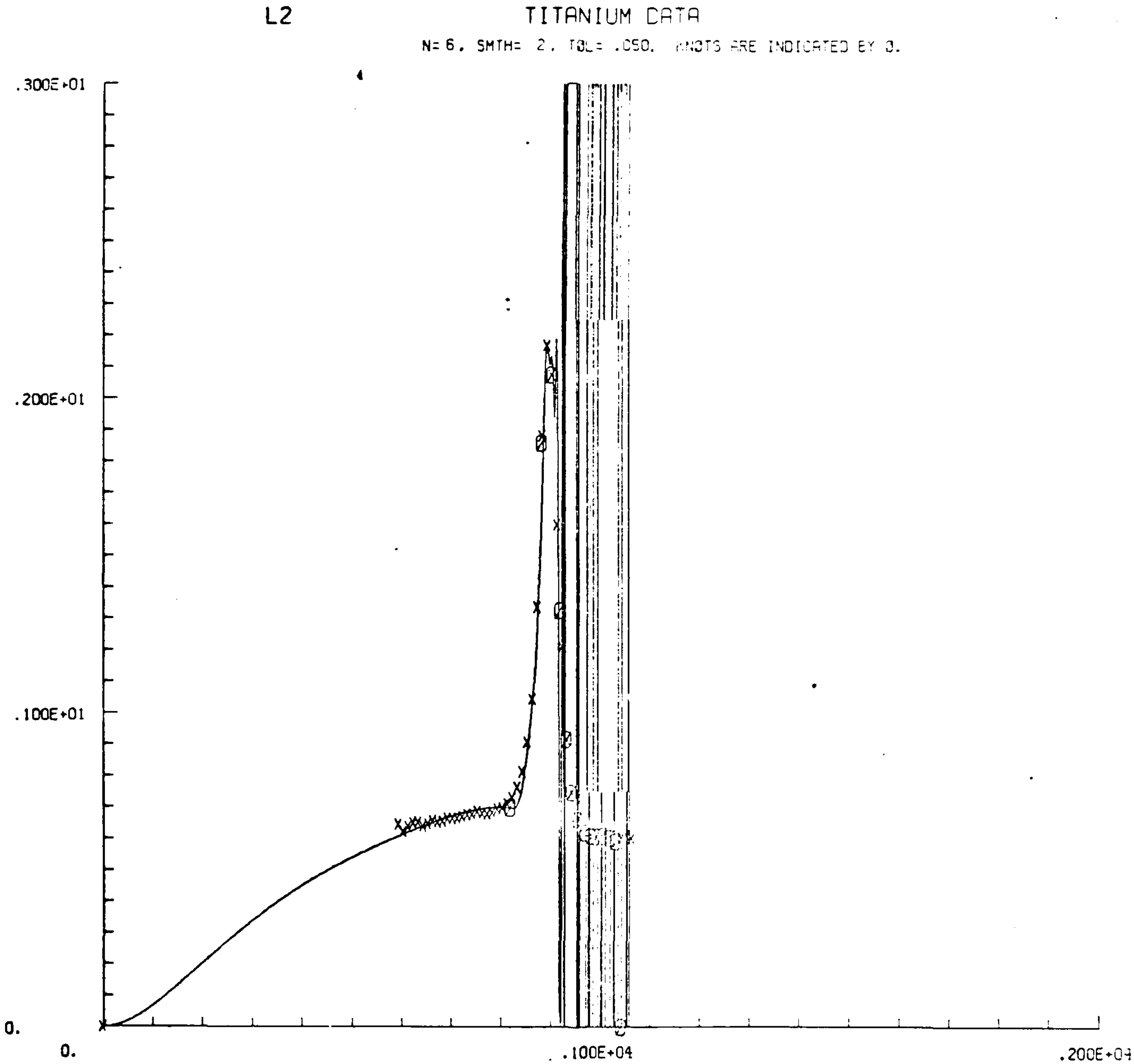

L2 TITANIUM DATA
N= 6. SMTH= 2, TØL= .050, KNØTS ARE INDICATED BY Ø.
.300E+01
.200E+01
.100E+01
0.
0.
.100E+04
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L2 APPRØX. ØPERATØR

Figure 15



PIECEWISE POLYNOMIAL APPROX. USING (DISCRETE) L2 APPROX. OPERATOR

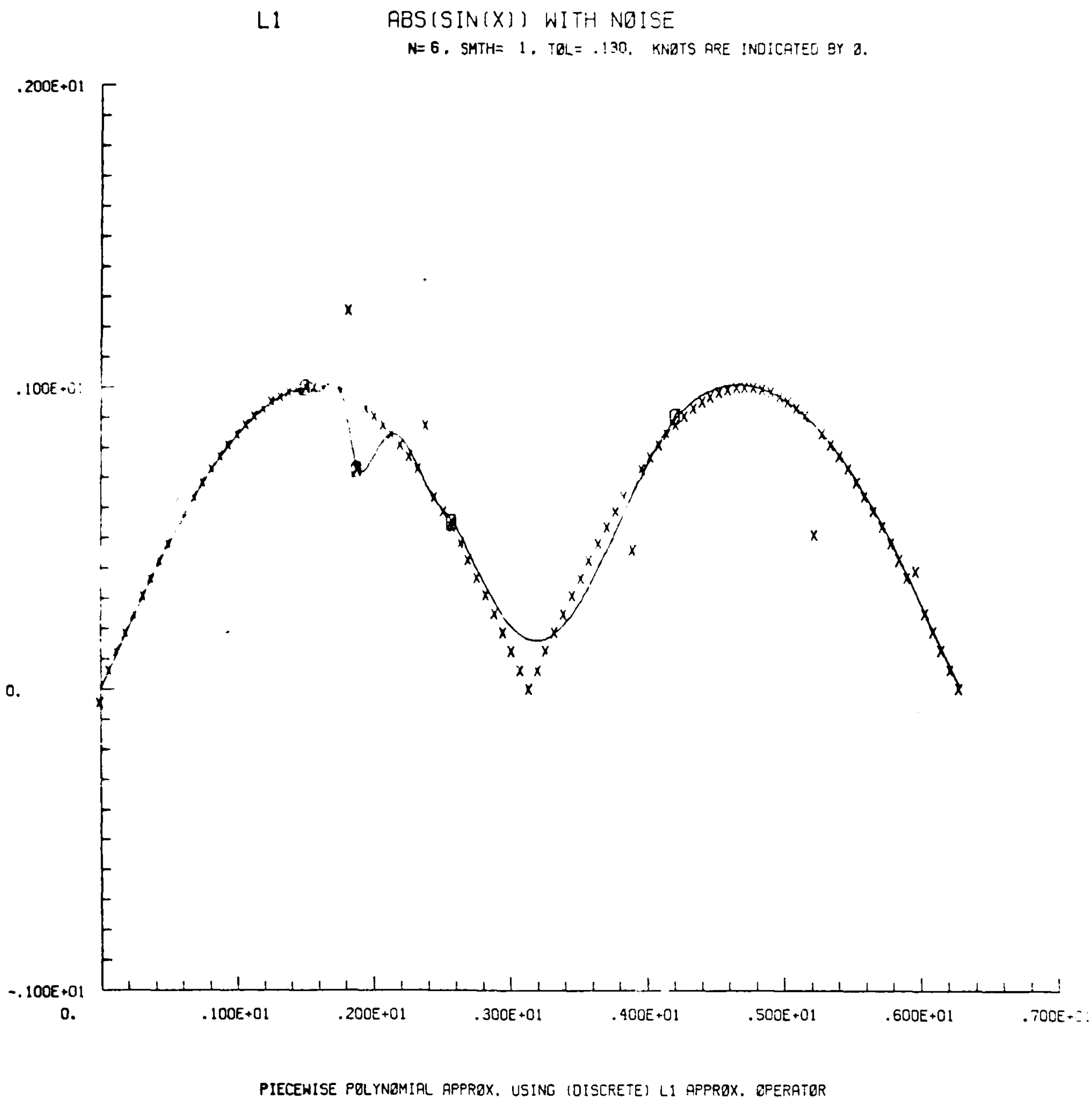

L1 ABS(SIN(X)) WITH NØISE
N= 6, SMTH= 1, TØL= .130, KNØTS ARE INDICATED BY Ø.
.200E+01
.100E+01
0.
-.100E+01
0.
.100E+01
.200E+01
.300E+01
.400E+01
.500E+01
.600E+01
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L1 APPRØX. ØPERATØR

Figure 17

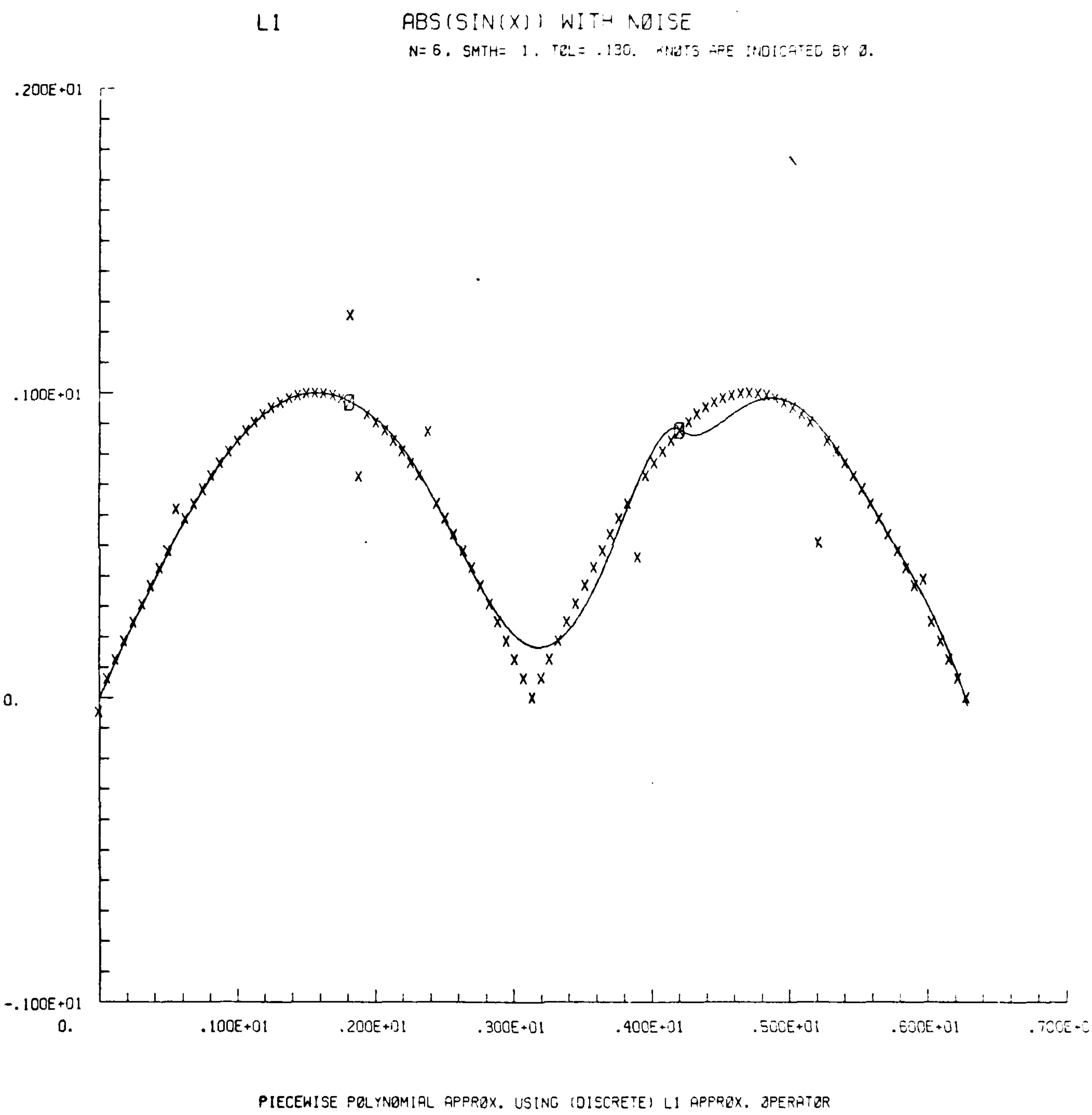
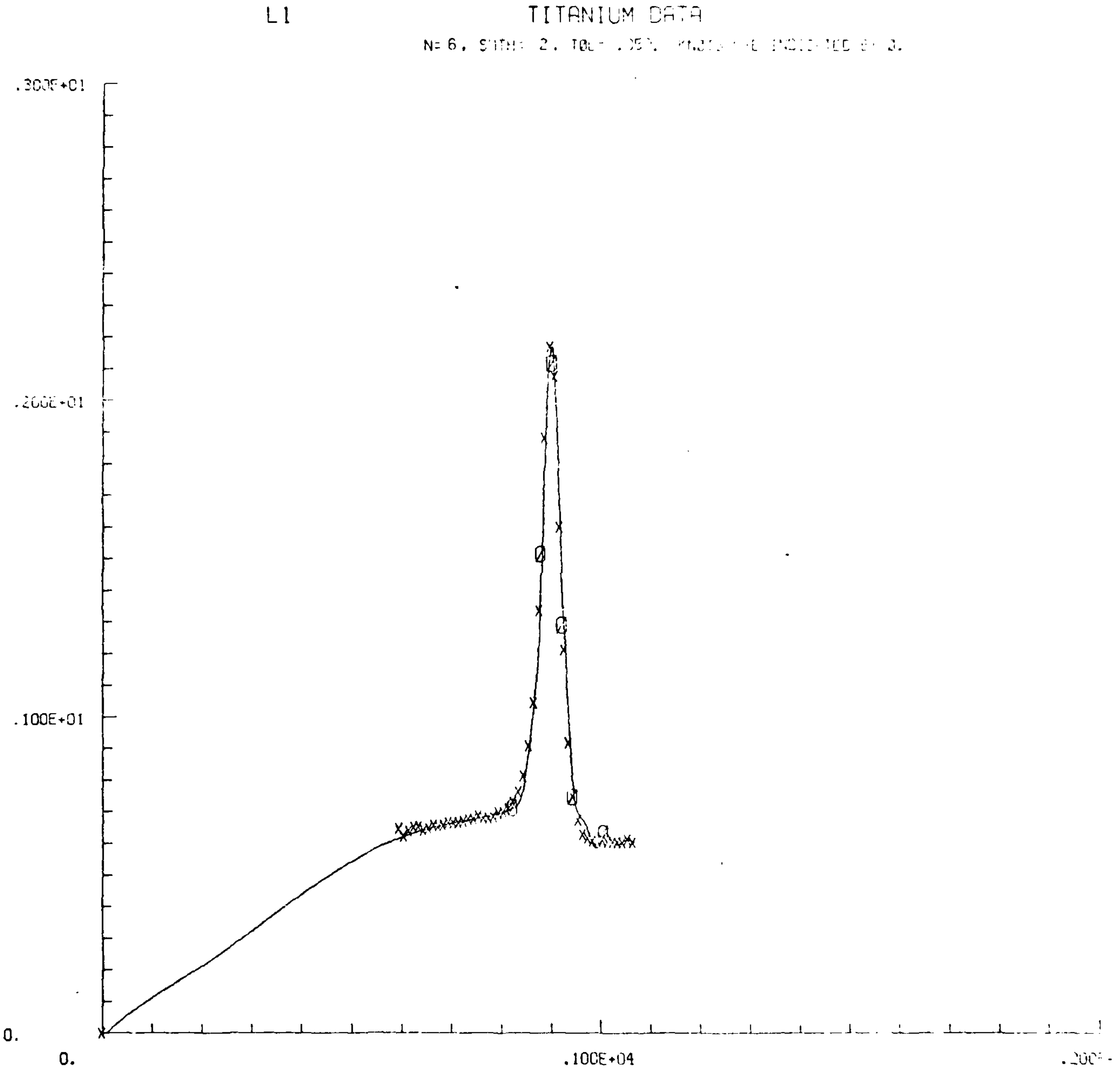
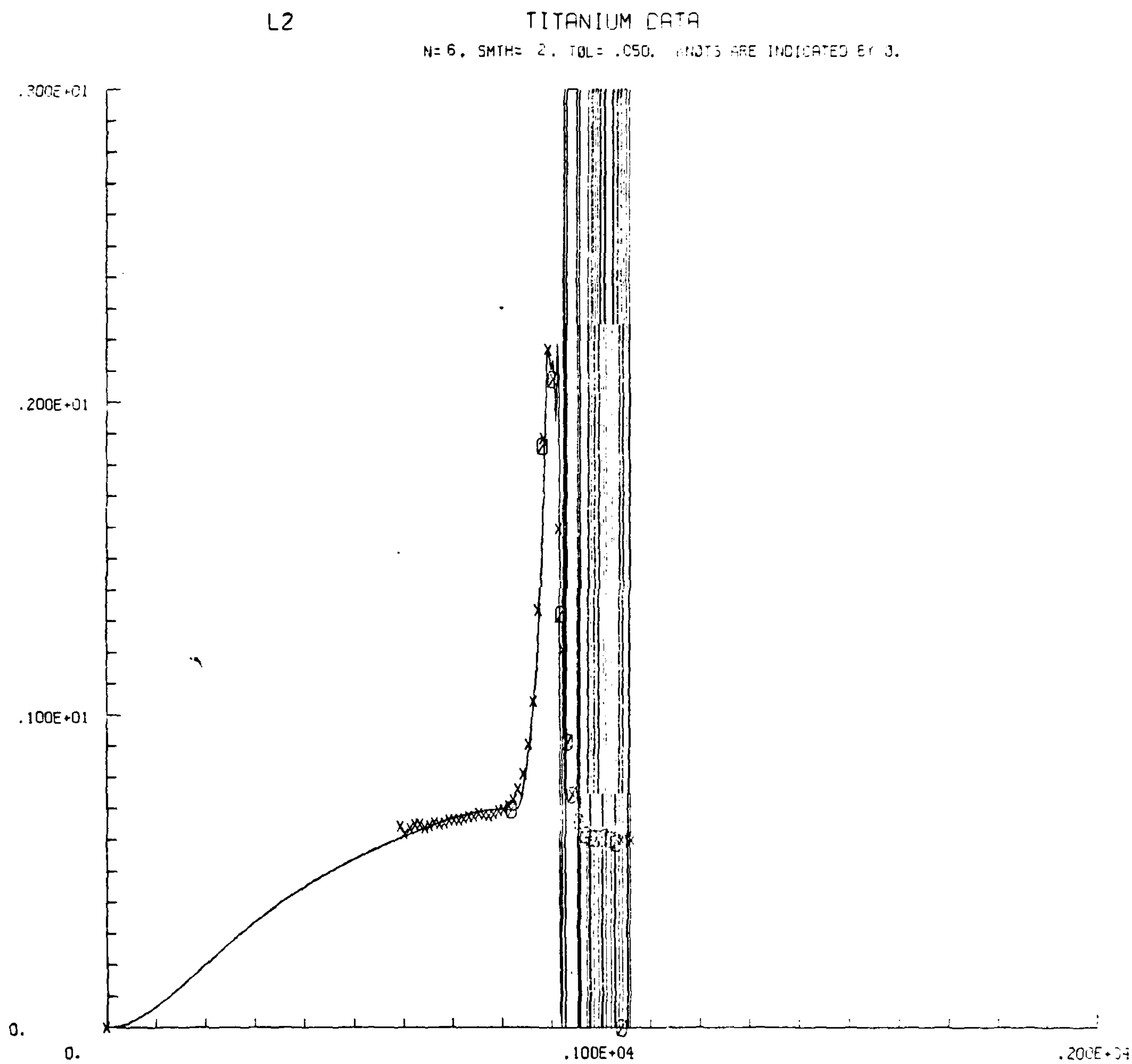

L1
ABS(SIN(X)) WITH NØISE
N= 6, SMTH= 1, TØL= .130. KNØTS ARE INDICATED BY Ø.
.200E+01
.100E+01
0.
-.100E+01
0.
.100E+01
.200E+01
.300E+01
.400E+01
.500E+01
.600E+01
.700E+01
PIECEWISE PØLYNØMIAL APPRØX. USING (DISCRETE) L1 APPRØX. ØPERATØR

Figure 18



PIECEWISE PØLYNOMIAL APPRØX. USING (DISCRETE) L1 APPRØX. ØPERATØR

Figure 19



PIECEWISE POLYNOMIAL APPROX. USING (DISCRETE) L2 APPROX. OPERATOR

REFERENCES

[1] P. G. Avila and G. D. Taylor, "Adaptive ℓ_1-ℓ_2 Curve Fitting", International Journal for Numerical Methods in Engineering, 14 (1979), 871-903.

[2] Thomas E. Spaulding, "Numerical Testing of Adaptive Curve Fitting Routines", M.S. Thesis, Colorado State University, 1979.